Chapter 8
X.25 Configuration and Management

8

This chapter describes how to configure connections through X.25 networks, including Level 2 Link Access Procedure–Balanced (LAPB) connections, and connections to a packet assembly/disassembly (PAD) facility. You will find the following information in this chapter:

- An overview of the X.25 standard.
- Configuration procedures for the X.25 Level 2 and Level 3 parameters, PAD connections compliant with the X.3 and X.29 standards, and X.29 access lists.
- EXEC commands for monitoring X.25 traffic and protocol specifications.

To configure X.25, you must be in the configuration command collection mode. To enter this mode, type the EXEC command **configure** at the EXEC prompt. Once in the collection mode, start configuring the interface by entering the **interface** command.

A command summary is included at the end of the chapter.

Cisco's Implementation of X.25

The software for the Cisco terminal server and protocol translator products supports the 1980 and 1984 Recommendation X.25, published by the French International Telegraph and Telephone Consultative Committee (CCITT). The X.25 standards specify connections between data terminal equipment (DTE) and data communications equipment (DCE). The Defense Data Network (DDN) and the International Standards Organization (ISO) specify the X.25 protocol for computer communications. Many public and private networks also use Recommendation X.25 as their interface technology.

The X.25 standards provide a telephone network model for computer data communications. To start data communications, one computer system calls another to request a communications session. The called computer system can accept or refuse the call. If the called system accepts the call, the two computer systems can begin transferring data in both directions; either system can terminate the call.

The X.25 standards also provide recommendations for the Data Link layer (OSI Level 2), to facilitate reliable interchange of data between devices across a link. The Cisco software supports Level 2 connections using X.25 Link Access Procedure–Balanced (LAPB) serial encapsulation methods.

Support for Terminal Connections and Protocol Translation

Cisco Systems terminal servers use the CCITT Recommendation X.25 for transferring raw data over X.25 networks. Cisco Systems terminal servers with protocol translation also support use of X.25 as a transport mechanism for Internet packets. In addition, the terminal server supports X.3 and X.29-based PAD connections. This allows the Cisco terminal servers to connect to an X.25 Public Data Network (PDN). Like Cisco routers, this X.25 connection allows transport of the TCP/IP packets across the X.25 packet switching network.

Cisco routers do not support an X.25 PAD function, so they are unable to communicate with hosts directly connected to the X.25 PDN. The routers encapsulate TCP/IP packets in X.25 packets for transfer over a packet switching network. These packets can be received by Cisco routers, TRouters, terminal servers, or other Cisco protocol translators. Only Cisco's protocol translators include PAD capabilities. However, other Cisco servers can communicate with a Cisco protocol translator using the Telnet or LAT protocols. The protocol translator can translate this into X.25, which then would be able to communicate with the X.25 host. The protocol translators support all PAD standards (X.28, X.29, and X.3). The connection to the packet switching network is through a synchronous line.

X.25 Configuration Overview

The Cisco X.25 software provides support for Level 2 LAPB connections, X.25 Level 3 connections, and with protocol translation, PAD connections. Summaries of the configuration procedure for each connection type are described in the following sections.

Configuring LAPB Connections

LAPB connections are supported by encapsulating datagrams so they may be transmitted over serial interfaces. The basic LAPB configuration procedure follows:

Step 1: Select the LAPB operation—either DTE or DCE—and then configure the interface with the appropriate LAPB encapsulation command.

Step 2: Specify the LAPB parameter settings. These settings are determined by the network and service over which the datagrams will be transmitted. Table 1-1 summarizes the LAPB parameters.

The commands for these tasks are described in the section "Configuring X.25 Level 2 (LAPB)." This section includes information about troubleshooting LAPB connections.

Configuring X.25 Connections

The basic X.25 configuration procedure follows:

Step 1: Assign the X.25 address to the interface. Obtain the X.25 address from the X.25 service provider.

Step 2: Select the X.25 operation—either DTE or DCE—and then configure the interface with the appropriate X.25 encapsulation command.

Step 3: Specify the X.25 Level 3 parameter settings. The default values provided by the software are sufficient for most X.25 networks; however, some parameters may need to be configured, depending on the network.

Optionally, protocol translation can be configured over multiple interfaces using the **x25 route** command. You can create tables by which to map IP datagrams to X.25 datagrams and thereby transmit IP data through an X.25 network. This is done using the **x25 map** command.

The commands for these tasks are described in the sections "Setting the X.25 Interface Address," "Configuring X.25 DTE or DCE Operation," "Address Mapping," and "Configuring X.25 Parameters" in this chapter. If you are connecting to a DDN X.25 network, see the section "Configuring DDN X.25." Commands for maintaining, monitoring, and troubleshooting the X.25 connections follow these sections.

Configuring PAD/X.29

Connections to a PAD are made using the EXEC commands **pad**, **resume**, and **x3**. You can configure PAD parameter profiles that can be used to set PAD parameters by other commands, and access lists to control X.25 network access. Both features make use of the message fields defined in Recommendation X.29, which describes procedures for exchanging data between two PADS or a PAD and a DTE. These tasks are described in the sections "Configuring X.29 Access Lists" and "Creating an X.29 Profile" in this chapter. Making PAD connections is described in the chapter "Terminal Server User Commands."

Configuring X.25 Level 2 (LAPB)

The X.25 standards distinguish between two types of X.25 hosts: data terminal equipment (DTE), and data communications equipment (DCE). At Level 2, or the Data Link level in the OSI model, X.25 provides the Link Access Procedure–Balanced (LAPB) to provide for orderly and reliable exchange of data between a DTE and a DCE.

Using LAPB under noisy conditions can result in greater throughput than HDLC encapsulation. When LAPB detects a damaged frame, the terminal server immediately retransmits the frame instead of waiting for host timers to expire. This behavior is good only if the host timers are relatively slow. In the case of quickly expiring host timers, however, you will discover that LAPB is spending much of its time retransmitting host transmissions.

However, if the line is not noisy, the lower overhead of HDLC encapsulation is more efficient than LAPB. When using long delay satellite links, for example, the lock-step behavior of LAPB makes HDLC encapsulation the better choice.

It is possible to only use LAPB as a serial encapsulation method. This can be done using a leased serial line. You must use one of the X.25 packet-level encapsulations when attaching to an X.25 network. A terminal server or protocol translator using LAPB encapsulation can act as a DTE or DCE device at the protocol level.

Configuring LAPB Encapsulation

To run datagrams over a serial interface using the LAPB encapsulation, use the interface subcommand:

encapsulation {lapb|lapb-dce}

The keyword **lapb** sets DTE operation; the keyword **lapb-dce** sets DCE operation. One end of the link must be DTE and the other must be DCE.

Sample Configuration of LAPB Encapsulation

In the following example of LAPB encapsulation configuration, the frame size (N1), window size (K), hold timer (TH), and maximum retransmission (N2) parameters retain their default values. The **encapsulation** subcommand sets DCE operation for IP packets only, and the **lapb t1** subcommand sets the retransmission timer to 4,000 milliseconds (4 seconds).

Example:

```
interface serial 3
encapsulation lapb-dce
lapb t1 4000
```

For more information on LAPB parameters, see the next section, "Setting the X.25 Level 2 (LAPB) Parameters."

Setting the X.25 Level 2 (LAPB) Parameters

X.25 Level 2, or LAPB (Link Access Procedure–Balanced), is a data encapsulation protocol that operates at Level 2 (the data link level) of the OSI reference model. LAPB specifies methods for exchanging data (in units called *frames*), detecting out-of-sequence or missing frames, retransmitting frames, and acknowledging frames.

LAPB parameters are set with the **lapb** interface subcommand. The interface must be running with either a LAPB or X.25 encapsulation method specified by the **encapsulation** interface subcommand. The command syntax follows:

lapb *parameter value*

This subcommand takes two required arguments, *parameter* and *value.*

The argument *parameter* is one of several keywords described in the following text, and the argument *value* is a decimal number representing a period of time, a bit count, or a frame count, depending on the parameter. Table 1-1 summarizes the LAPB parameters.

Table 1-1 LAPB Parameters

Parameter	Value	Value Range	Default
k	*frames*	1-7	7
n1	*bits*	1-16384	12000
n2	*times*	1-255	20
t1	*milliseconds*	1-64000	3000

Note: The LAPB "th" value is not included as a configurable parameter; the value is always 0.

Setting the Retransmission Timer

The retransmission timer determines how long a transmitted frame can remain unacknowledged before the terminal server polls for an acknowledgment. To set the limit for the retransmission timer (the LAPB T1 parameter), use the **lapb t1** subcommand. The command syntax follows:

lapb t1 *milliseconds*

The argument *milliseconds* is the number of milliseconds from 1 through 64,000. The default value is 3,000 milliseconds.

For X.25 networks, the terminal server retransmission timer setting should match that of the network. Mismatched retransmission timers can cause excessive retransmissions and an effective loss of bandwidth.

For leased-line circuits, the retransmission timer setting is critical. The timer setting must be large enough to permit a maximum-sized frame to complete one round trip on the link. If the timer setting is too small, the terminal server will poll before the acknowledgment frame can return, which results in an effective loss of bandwidth. If the timer setting is too large, the terminal server waits longer than necessary before requesting an acknowledgment, which also reduces bandwidth.

To determine an optimal value for the retransmission timer, use the privileged EXEC command **ping** to measure the round-trip time of a maximum-sized frame on the link. Multiply this time by a safety factor that takes into account the speed of the link, the link quality, and the distance. A typical safety factor is 1.5. Choosing a larger safety factor can result in slower data transfer if the line is noisy. However, this disadvantage is minor compared to the excessive retransmissions and effective bandwidth reduction caused by a timer setting that is too small.

Setting Frame Parameters

To specify the maximum number of bits a frame can hold, use the **lapb n1** interface subcommand:

lapb n1 *bits*

The **n1** keyword specifies the maximum number of bits (N1) a frame can hold. The argument *bits* is the number of bits from 1 through 12000, and must be a multiple of eight. The default value is 12000 bits (1500 bytes).

When connecting to an X.25 network, use the N1 parameter value set by the network administrator, which is the maximum size of an X.25 packet. When using LAPB over leased lines, the N1 parameter should be eight times the MTU.

To specify the maximum number of times an acknowledgment frame can be retransmitted, use the **lapb n2** interface subcommand:

lapb n2 *retries*

The argument *retries* is the retransmission count from 1 through 255. The default value is 20 retransmissions.

To specify the maximum permissible number of outstanding frames, called the "window size," use the **lapb k** interface subcommand:

lapb k *window-size*

The argument *window-size* is a packet count from 1 to 7. The default value is 7 packets.

Example:

The following configuration commands change the LAPB window size (the K parameter) to 3 packets:

```
interface serial 0
lapb k 3
```

To adjust the Level 3 (packet level) parameters of an X.25 interface, use the **x25** subcommand of the **interface** configuration command.

Monitoring and Troubleshooting LAPB

To display operation statistics for an interface using LAPB encapsulation, use the EXEC command **show interfaces.**

The following example output shows the state of the LAPB protocol, the current parameter settings, and a count of the different types of frames. Each frame count is displayed in the form sent/received.

```
LAPB state is DISCONNECT, T1 3000, N1 12000, N2 20, K 7,TH 3000
IFRAMEs 12/28 RNRs 0/1 REJs 13/1 SABMs 1/13 FRMRs 3/0 DISCs 0/11
```

For a description of the variable names in the **show interface** output, see the X.25 recommendation.

To debug LAPB problems, you must have a good understanding of the X.25 recommendation.

To enable the logging of all packets received and generated, use the privileged EXEC command **debug lapb**. Note that this command slows down processing considerably on heavily loaded links. The following shows example output:

```
Serial0: LAPB 0 CONNECT (5) IFRAME 0 1
Serial0: LAPB I CONNECT (2) RR 1 (R)
Serial0: LAPB I CONNECT (5) IFRAME P 2 1 (C)
Serial0: LAPB 0 REJSENT (2) REJ P/F 1
Serial0: LAPB I REJSENT (2) DM F (C)
Serial0: LAPB I DISCONNECT (2) SABM (C)
Serial0: LAPB 0 CONNECT (2) UA
.
.
.
Serial0: LAPB T SABMSENT 357964 0
Serial0: LAPB 0 SABMSENT (2) SABM P
```

In the example output, each line represents a LAPB frame entering or exiting the terminal server. The first field shows the interface type and unit number of the interface reporting the frame event. The second field is the protocol that provided the information.

The third field is `I`, `O`, or `T` for "frame input," "frame output," or "T1 timer expired," respectively. The fourth field indicates the state of the protocol when the frame event occurred. In a timer event, the state name is followed by the current timer value and the number of retransmissions.

In a packet input or output event, the state name is followed by the size of the frame in bytes (in parentheses) and the frame type name. The next field is an optional indicator: `P/F`, `P`, or `F`, which stand for "Poll/Final," "Poll," and "Final," respectively. For IFRAME frames only, the next two numbers are the receive and send sequence numbers, respectively. For RR, RNR, and REJ frames, the next number is the receive sequence number. For FRMR frames, the next three numbers are three bytes of error data. The last optional indicator is `(C)` for "command" or `(R)` for "response."

Setting the X.25 Interface Address

To set the X.121 address of a particular network interface, use the **x25 address** interface subcommand:

x25 address *X.121-address*

The argument *X.121-address* is a variable-length X.121 address. The address is assigned by the X.25 network service provider.

Example:

The following configuration commands set the X.121 address for the interface:

```
interface serial 0
x25 address 00000123005
```

The value must match that assigned by the X.25 network.

To display the current X.25 parameter settings, use the EXEC command **show interfaces**. Refer to the "Monitoring X.25" section later in this chapter for more information.

Configuring X.25 DTE or DCE Operation

This section describes the different encapsulation methods for selecting device operation. Methods of encapsulation for DDN networks are described in the section "Configuring DDN X.25" later in this chapter.

A terminal server using X.25 Level 3 encapsulation can act as a DTE or DCE device on general X.25 networks.

To set X.25 DTE operation, use the **encapsulation x25** interface subcommand:

encapsulation x25

To set X.25 DCE operation, use the **encapsulation x25-dce** interface subcommand:

encapsulation x25-dce

Default serial encapsulation is HDLC. You must choose one of these X.25 encapsulation methods.

Connecting to Multiple X.25 Interfaces

The Cisco X.25 software allows you to connect to multiple X.25 interfaces. Use the global configuration command **x25 route** to make these connections. Configure each of the X.25 interfaces as appropriate for the network it is connected to, and then use the **x25 route** command to build the tables of network addresses.The full syntax and variations of the **x25 route** command follow.

x25 route [# *position***]***x121-pattern* **[cud** *pattern***] interface** *interface-name*
no x25 route [# *position***]***x121-pattern* **[cud** *pattern***] interface** *interface-name*

x25 route [# *position***]***x121-pattern* **[cud** *pattern***] ip** *ip-address*
no x25 route [# *position***]***x121-pattern* **[cud** *pattern***] ip** *ip-address*

x25 route [# *position***]***x121-pattern* **[cud** *pattern***] alias** *interface-name*
no x25 route [# *position***]***x121-pattern* **[cud** *pattern***] alias** *interface-name*

The order in which entries are specified is significant; the list is scanned linearly for the first match. The optional positional parameter (# followed by an integer) can be used to designate after which existing entry to insert or delete the new entry. If no positional parameter is given, the entry is appended to the end of the routing table.

The argument *x121-pattern* is the called X.121 address and is required. Optional Call User Data corresponding to this X.121 address can be specified as a printable ASCII string. The Call User Data field (**cud** *pattern*) specifies the data after the protocol identification field, which is four bytes. Both the X.121 address and Call User Data can be written using UNIX-style regular expressions.

See Table 1-2 and Table 1-3 for a summary of pattern and character matching symbols and their use. A more complete description of the pattern matching characters is found in the appendix "Pattern Matching."

The **interface** keyword and *interface-name* argument define the interface.

The **ip** keyword and *ip-address* argument specify an IP address.

The **alias** keyword and *interface-name* argument permit a way for other X.121 addresses to be treated as local address. An alias accepts calls for the terminal server.

Enter the **no x25 route** command with the appropriate arguments and keywords to remove an entry from the table. Use the **show x25 route** command to display the X.25 routing table. The interface routes will show up after any routes used for translation commands. Since the interface routes are expected to be less specific, they should come last. This is done automatically.

Note: Load sharing to a single X.25 network is not currently supported.

Table 1-2 Pattern Matching

Pattern	Description
\0	Replaces the entire original address.
\1...9	Replaces strings that match the first through ninth parenthesized part of the X.121 address.
*	Matches 0 or more sequences of the regular expressions.
+	Matches 1 or more sequences of the regular expressions.
?	Matches the regular expression of the null string.

Table 1-3 Character Matching

Character	Description
^	Matches the null string at the beginning of the input string.
$	Matches the null string at the end of the input string.
char	Matches *char.*
.	Matches any single character.

Example:

The following example illustrates how to use three interfaces to connect to three different X.25 services. The **x25 address** command assigns the X.25 address to the interface, then the table is generated using the **x25 route** command. Notice that regular expression pattern matching characters are used to match just the initial portion of the complete X.25 addresses. Finally, protocol translation is defined as IP packets encapsulated into X.25 data using the **translate** command.

```
! This PT has three X.25 interfaces.  One is connected to infonet, one
! to tymnet, and one to an internal private X.25 network
!
interface serial 0
description InfoNet Connection
x25 address 3107012332004
!
interface serial 1
description Tymnet connection
x25 address 3106012332001
!
interface serial 2
description Internal X.25 net
x25 address 111110000001
x25 win 7
x25 wout 7
x25 ips 512
x25 ops 512
x25 route ^3107 interface serial 0
x25 route ^3106 interface serial 1
x25 route ^1111 interface serial 2
!
translate tcp infonet-noc x25 31070
translate tcp tymnet-noc x25 3106011
translate tcp x25noc x25 11110000022
```

Address Mapping

The Cisco Systems terminal servers can use three methods to map Internet addresses to X.121 addresses used by X.25.

Note: Mapping of IP addresses to X.121 addresses is necessary only when running IP over X.25.

A terminal server set up for DDN service uses the dynamic mapping technique specified in Appendix A of the *DDN X.25 Host Interface Specification*, available from the DDN Network Information Center (NIC). This technique has two severe limitations: it applies only to Class A Internet addresses, and it ignores the third octet of the Internet address; this technique works well for the DDN, but not for any other network.

You can establish the X.121 address of a particular network interface using the **x25 address** interface subcommand. If you do not specify the address, the terminal server uses the DDN mapping technique to obtain the X.121 address of an interface.This command is described in the section "Setting the X.25 Level 3 Parameters" in this chapter.

You can set up the Internet-to-X.121 address mapping for the host using the **x25 map** interface subcommand. Because no defined protocol can dynamically determine such mappings, you must enter a mapping for each host with which the terminal server will exchange traffic. The full command syntax follows:

x25 map *protocol-keyword protocol-address X.121-address* **[***option1…* **[***option***]]**
no x25 map *protocol-keyword protocol-address X.121-address*

For the terminal server, the argument *protocol-keyword* can only be **ip**. The *address* arguments specify the network-protocol-to-X.121 mapping.

The *option* arguments add certain features to the mapping specified, and can be any of the following, up to six. (They must be specified in the order listed.)

- **reverse**—Specifies reverse charging for outgoing calls.
- **accept-reverse**—Causes the terminal server to accept incoming reverse-charged calls. If this option is not present, the terminal server clears reverse charge calls.
- **broadcast**—Causes the terminal server to direct any broadcasts sent through this interface to the specified X.121 address.
- **cug** *number*—Specifies a Closed User Group number (from 1 to 99) for the mapping in the outgoing call.
- **nvc** *count*—Sets the number of virtual circuits (VCs) for this map/host. The default *count* is the **x25 nvc** setting of the interface. A maximum number of eight VCs can be configured for a single map/host.
- **packetsize** *in-size out-size*—Specifies input packet size *in-size* and output packet size *out-size* for the mapping in the outgoing call.
- **windowsize** *in-size out-size*—Specifies input window size *in-size* and output window size *out-size* for the mapping in the outgoing call.
- **throughput** *in out*—Requests the amount of bandwidth through the X.25 network.
- **modulo** *size*—Specifies the maximum window size for this map. The argument *size* permits windows of 8 or 128 on the same interface.

- **transit-delay** *number*—Specifies the transit delay value in milliseconds (0 to 65334) for the mapping of outgoing calls, for networks that support transit delay.
- **nuid** *username password*—Specifies that a network ID facility be sent in the outgoing call with the specified user name and password.

To retract a network-protocol-to-X.121 mapping, use the **no x25 map** command with the appropriate network protocol and X.121 address arguments.

You can define a static host-name-to-address mapping in the host cache using the **x25 host** subcommand. The command syntax follows:

x25 host *name 121-address* **[cud call-user-data]**
no x25 host *name*

The argument *name* is the host name. The keyword **cud call-user-data** sets the call user data field in the X.25 call request packet. This keyword is optional.

X.25 TCP Header Compression

The implementation of Compressed TCP over X.25 uses another virtual circuit (VC) to pass the compressed packets. The noncompressed packets use one VC and the compressed packets take another VC.

Use this interface subcommand to implement head compression on IP packets:

ip tcp header-compression [passive]
no ip tcp header-compression [passive]

Use the following interface subcommand to make the X.25 calls complete for compressed packets:

x25 map compressedtcp *ip-address x.121-address* **[***options***]**
no x25 map compressedtcp *ip-address x.121-address*

The argument *ip-address* is the IP address and *x.121-address* is the X.121 address. The *options* arguments are the same options as those for the **x25 map** command; see above. The Call User Data of compressed TCP calls is the single byte 0xd8.

Configuring DDN X.25

The DDN X.25 protocol has two versions: Basic Service and Standard Service. Using the DDN X.25 Basic Service, network devices send raw X.25 data across the DDN and assume no structure within the data portion of the X.25 packet. Basic Service users can interoperate only with other Basic Service users.

DDN X.25 Standard Service requires that the X.25 data packets carry IP datagrams. Because the DDN Packet Switch Nodes (PSNs) can extract the Internet packet from within the X.25 packet, they can pass data to either an 1822-speaking-host or to another Standard Service host. Thus, hosts using the older 1822 network interface can interoperate with hosts using Standard Service.

The DDN X.25 Standard is the required protocol for use with DDN PSNs. The Defense Communications Agency (DCA) has certified the Cisco Systems DDN X.25 Standard implementation for attachment to the Defense Data Network.

For situations requiring a high degree of security, the Defense Data Network Blacker Front-End Encryption (BFE) device is supported. If the router is attached to such a device, the **bfex25** keyword must be used with the **encapsulation** subcommand:

encapsulation bfex25

This encapsulation provides a mapping from Class A IP addresses to the type of X.121 addresses expected by the BFE encryption device.

Configuring DDN X.25 DTE or DCE Operation

A terminal server using the DDN X.25 Basic Service can act as a DTE or DCE device. To set operation type, use this interface subcommand:

encapsulation {ddnx25|ddnx25-dce}

These keywords cause the terminal server to specify the Standard Service facility in the Call Request packet, which notifies the PSNs to use Standard Service.

Using Standard Service, the DDN can provide better service for virtual circuits with higher precedence values. If the terminal server receives an Internet packet with a nonzero Internet precedence field, it opens a new virtual circuit and sets the precedence facility request to the DDN-specified precedence mapping in the Call Request packet. Different virtual circuits are maintained based on the precedence mapping values and the permitted number of virtual circuits.

Defining the Type of Service Field

Normally the X.25 parameters of a DDN connection are configured using the **x25** and **lapb** interface subcommand described in the section "Configuring the X.25 Parameters;" however, there are a few DDN-specific subcommands.

The Cisco X.25 implementation allows you to enable or disable the ability to open a new virtual circuit based on the IP Type of Service (TOS) field.

To do this, use the **x25 ip-precedence** interface subcommand. The full syntax of this command follows:

x25 ip-precedence
no x25 ip-precedence

By default, Cisco terminal servers open one virtual circuit for each type of service. There is a problem associated with this in that some hosts send nonstandard data in the TOS field, thus causing multiple, wasteful virtual circuits to be created. The command **no x25 ip-precedence** causes the TOS field to be ignored when opening virtual circuits.

Using the HDH Protocol

The HDH protocol (also known as the HDLC Distant Host or 1822-J protocol) provides a method for running the 1822 protocol over synchronous serial lines instead of over special-purpose 1822 hardware. HDH packets consist of 1822-LH/DH leaders and data encapsulated in LAPB (X.25 Level 2) format. The HDH hardware is on the Multiport Communications Interface (MCI) card. Strictly speaking, HDH is not X.25, however, it is still commonly used for attaching to the Defense Data Network.

The terminal server supports both the packet and message modes of HDH. It is enabled with the **hdh** interface subcommand with the appropriate keyword. The syntax follows:

hdh {packet|message}

The **packet** keyword specifies the packet mode; the **message** keyword specifies the message mode.

To enable the HDH protocol, use the interface subcommand:

encapsulation hdh

To enable logging of HDH transactions, use the privileged EXEC command **debug hdh.** To enable logging of the underlying LAPB protocol transactions, use the privileged EXEC command **debug lapb.**

To display information about HDH, use the EXEC command **show imp-hosts.** This command displays information about the hosts with which the network server has communication during the past five minutes via its CSC-A (1822-LH/DH) interfaces or serial interface using HDH (1822-J) encapsulation.

Use the EXEC command **debug psn** to log information about the Packet Switch Node. This command enables logging of noteworthy Packet Switch Node (PSN) events for DDN network servers that are equipped with CSC-A (1822-LH/DH) interfaces or serial interfaces using HDH (1822-J) encapsulation.

The **debug psn-events** EXEC command enables logging of a subset of the PSN and 1822 debugging messages.

Configuring X.25 Parameters

The Cisco terminal server software provides subcommands to configure the standard Level 2 and Level 3 X.25 parameters and user facilities.

This section discuses the commands and procedures needed to set these parameters including interface addresses, virtual circuit channel sequences, and addresses.

Setting the X.25 Level 3 Parameters

This section describes the keywords for the **x25** subcommand of the interface configuration command, used to adjust certain X.25 Level 3 parameters. These parameters must be adjusted to match the values used by the X.25 network. It is common for networks to require values different from the Cisco defaults.

Note: If you connect a terminal server to an X.25 network, use the parameters set by the network administrator. Also, note that the X.25 Level 2 parameters described in "Setting the X.25 Level 2 (LAPB) Parameters" earlier in this chapter affect X.25 Level 3 operations.

Configuring the Virtual Circuit Channel Sequence

An important part of X.25 operation is the virtual circuit channel sequence. This sequence is a range of virtual circuit channel numbers broken into four groups (listed here in numerically increasing order):

1. Permanent virtual circuits
2. Incoming calls
3. Incoming and outgoing (two-way) calls
4. Outgoing calls

Several X.25 parameters determine the numerical ranges of the last three groups; the range for permanent virtual circuits falls numerically below the incoming call range.

X.25 communications devices use the virtual circuit channel sequence when allocating virtual circuits. When initiating a call, these devices must search for an available channel in the sequence in one of two ways. For outgoing calls (made by a DTE device), the search starts at the upper end of the outgoing call range and proceeds in the direction of decreasing channel numbers. The search continues until the device finds an available channel or reaches the lower limit of the two-way call range.

For incoming calls (handled by a DCE device), the search for which channel numbers to allocate starts at the lower end of the incoming call range, and proceeds in the direction of increasing channel numbers. The search continues until the device finds an available channel or reaches the upper limit of the two-way call range. The DTE and DCE devices fail to find an available channel only if the overall sequence range is very small or when all of the channels are in use.

To set the upper and lower limit parameters of the channel ranges, use the **x25** subcommand keywords listed in Table 1-4. Each keyword takes a channel number as its argument. Note that the values for these parameters must be the same on both ends of an X.25 link.

The default lower limit for all channel ranges on the terminal server is 1; the default upper limit for all ranges is 1024. These defaults reflect a simple approach to allocating the channel ranges: assign the same low value to all lower limits, and assign the same high value to all upper limits. This approach causes the three ranges to overlap and become one large range.

Table 1-4 Range Limit Keywords for the Virtual Circuit Channel Sequence

Keyword	Limit Type	Range	Default
lic	Lower limit, incoming call range	1-4095	1
hic	Upper limit, incoming call range	1-4095	1024
ltc	Lower limit, two-way call range	1-4095	1
htc	Upper limit, two-way call range	1-4095	1024
loc	Lower limit, outgoing call range	1-4095	1
hoc	Upper limit, outgoing call range	1-4095	1024

Setting the Highest Incoming Channel

The **hic** keyword sets the highest incoming channel (HIC).

x25 hic *channel*

The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

Setting the Highest Outgoing Channel

The **hoc** keyword sets the highest outgoing channel (HOC).

x25 hoc *channel*

The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

Setting the Highest Two-Way Channel

The **htc** keyword sets the highest two-way channel (HTC).

x25 htc *channel*

The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

Setting the Lowest Incoming Channel

The **lic** keyword sets the lowest incoming channel (LIC).

x25 lic *channel*

The argument *channel* is a channel number from 1 through 4095. The default value is 1.

Setting the Lowest Outgoing Channel

The **loc** keyword sets the lowest outgoing channel (LOC).

x25 loc *channel*

The argument *channel* is a channel number from 1 through 4095. The default value is 1.

Setting the Lowest Two-Way Channel

The **ltc** keyword sets the lowest two-way channel (LTC).

x25 ltc *channel*

The argument *channel* is a channel number from 1 through 4095. The default value is 1.

Example:

These commands set the following channels:

```
! set 01-20 incoming
x25 hic 20
! set 01-20 either incoming or outgoing
x25 htc 20
! set 01-20 outgoing
x25 hoc 20
```

Clearing Switched Virtual Circuits

The terminal server can clear a Switched Virtual Circuit (SVC) after a set period of inactivity. To set this period, use the **x25 idle** interface subcommand.

x25 idle *minutes*

The argument *minutes* is the number of minutes in the period. The default value is zero, which causes the terminal server to keep the SVC open indefinitely. Both calls originated and terminated by the terminal server are cleared.

To clear all virtual circuits at once, use the privileged EXEC command **clear x25-vc.** This command takes an interface type keyword (usually **serial**) and a unit number as arguments to identify the interface with which the virtual circuits are associated.

To clear a particular virtual circuit, the **clear x25-vc** command takes the two arguments described above and a logical circuit number (LCN) value as a third argument to specify the circuit.

Increasing the Number of VCs

To increase throughput across networks, you can establish up to eight switched virtual circuits to a host. To specify the maximum number of switched virtual circuits that can be open simultaneously to one host, use the **x25 nvc** interface subcommand:

x25 nvc *count*

The argument *count* is a circuit count from 1 to 8. The default is 1.

Configuring the Ignore VC Timer

Upon receiving a Clear Request for an outstanding Call Request, the X.25 support code immediately tries another Call Request, if it has more traffic to send. This can overrun some X.25 switches. To prevent this problem, use the **x25 hold-vc-timer** configuration command:

x25 hold-vc-timer *minutes*

The argument *minutes* is the number of minutes it takes to prevent calls from going to a previously failed destination. Incoming calls will still be accepted.

Configuring the X.25 Level 3 Retransmission Timers

The X.25 Level 3 retransmission timers determine how long the terminal server must wait before retransmitting various Call Request packets. You can set these timers independently using the **x25** subcommand keywords listed in Table 1-5. Each keyword requires a time value in seconds as its argument. The last column shows the default timer values, in seconds. Four of the timers apply to DTE devices, and the other four apply to DCE devices.

Table 1-5 Retransmission Timer Keywords and Defaults

Keyword (DTE/DCE)	Affected Request Packet	Time (seconds) (DTE/DCE)
t20/t10	Restart Request	180/60
t21/t11	Call Request	200/180
122/t12	Reset Request	180/60
t23/13	Clear Request	180/60

Setting the DTE Restart Request Retransmission Timer

The **t20** keyword sets the limit for the Restart Request retransmission timer (T20) on DTE devices.

x25 t20 *seconds*

The argument *seconds* is a time value in seconds. The default is 180 seconds.

Setting the DCE Restart Request Retransmission Timer

The **t10** keyword sets the limit for the Restart Request retransmission timer (T10) on DCE devices.

x25 t10 *seconds*

The argument *seconds* is a time value in seconds. The default value is 60 seconds.

Setting the DTE Call Request Retransmission Timer

The **t21** keyword sets the limit for the Call Request retransmission timer (T21) on DTE devices.

x25 t21 *seconds*

The argument *seconds* is a time value in seconds. The default value is 200 seconds.

Setting the DCE Call Request Retransmission Timer

The **t11** keyword sets the limit for the Call Request retransmission timer (T11) on DCE devices.

x25 t11 *seconds*

The argument *seconds* is a time value in seconds. The default value is 180 seconds.

Setting the DTE Reset Request Retransmission Timer

The **t22** keyword sets the limit for the Reset Request retransmission timer (T22) on DTE devices.

x25 t22 *seconds*

The argument *seconds* is a time value in seconds. The default value is 180 seconds.

Setting the DCE Reset Request Retransmission Timer

The **t12** keyword sets the limit for the Reset Request retransmission timer (T12) on DCE devices.

x25 t12 *seconds*

The argument *seconds* is a time value in seconds. The default value is 60 seconds.

Setting the DTE Clear Request Retransmission Timer

The **t23** keyword sets the limit for the Clear Request retransmission timer (T23) on DTE devices.

x25 t23 *seconds*

The argument *seconds* is a time value in seconds. The default value is 180 seconds.

Setting the DCE Clear Request Retransmission Timer

The **t13** keyword sets the limit for the Clear Request retransmission timer (T13) on DCE devices.

x25 t13 *seconds*

The argument *seconds* is a time value in seconds. The default value is 60 seconds.

Setting X.25 Packet Sizes

X.25 networks use maximum input and output packet sizes set by the network administration. You can set the terminal server input and output packet sizes to match those of the network with the **x25** subcommand keywords **ips** and **ops,** respectively:

x25 ips *bytes*
x25 ops *bytes*

The argument *bytes* is a byte count in the range of 128 through 4096. The default value is 128 bytes. Larger packet sizes are better, because smaller packets require more overhead processing.

Note: Set the **x25 ips** and **x25 ops** keywords to the same value unless your network supports asymmetry between input and output packets.

To send a packet larger than the X.25 packet size over an X.25 virtual circuit, a terminal server must break the packet into two or more X.25 packets with the M-bit ("more data" bit) set. The receiving device collects all packets with the M-bit set and reassembles them.

Setting the Flow Control Modulus

X.25 supports flow control with a sliding window sequence count. The window counter restarts at zero upon reaching the upper limit, which is called the *window modulus*. To set the window modulus, use the **x25 modulo** interface subcommand:

x25 modulo *modulus*

The argument *modulus* is either 8 or 128. The default value is 8. The value of the modulo parameter must agree with that of the device on the other end of the X.25 link.

Configuring Packet Acknowledgment

To specify upper limits on the number of outstanding unacknowledged packets, use the **x25** subcommand keywords **win** (for input window) and **wout** (for output window):

x25 win *packets*
x25 wout *packets*

The argument *packets* is a packet count. The packet count for **win** and **wout** can range from 1 to the window modulus. The default value is 2 packets.

The **x25 win** determines how many packets the terminal server can receive before sending an X.25 acknowledgment; the number is one less than **win.** The **wout** limit determines how many sent packets can remain unacknowledged before the terminal server uses a hold queue. To maintain high bandwidth utilization, assign these limits the largest number that the network allows.

Note: Set **win** and **wout** to the same value unless your network supports asymmetry between input and output window sizes.

You can instruct the terminal server to send acknowledgment packets when it is not busy sending other packets, even if the number of input packets has not reached the **win** count. This approach improves line responsiveness at the expense of bandwidth.

To enable this option, use the **x25 th** subcommand:

x25 th *delay*

The argument *delay* must be between zero and the input window size. A value of 1 sends one Receiver Ready acknowledgment per packet at all times. The default value is zero, which disables the delayed acknowledgment strategy.

The terminal server sends acknowledgment packets when the number of input packets reaches the count you specify, providing there are no other packets to send. For example, if you specify a count of 1, the terminal server can send an acknowledgment per input packet.

Setting the X.25 Level 3 Facilities

This section describes the **x25** keywords for setting the X.25 Level 3 Facilities.

Setting the Default Protocol

To set the default protocol, use the **x25 default** interface subcommands:

x25 default *protocol*
no x25 default *protocol*

The **default** keyword specifies the protocol assumed by the terminal server to interpret incoming calls with unknown Call User Data. The argument *protocol* is one of the following keywords:

- **ip**
- **pad**

There is no initial default value. If you do not use the **x25 default** subcommand, the terminal server clears any incoming calls with unknown Call User Data.

If you use this subcommand, the terminal server assumes that these calls use the specified protocol.

Use the **no x25 default** subcommand to remove the protocol specified.

This subcommand is used when an incoming call is received without identifying Call User Data. Normally, the call is cleared when this subcommand is used; the incoming call is assumed to contain the specified default protocol.

Suppressing the Calling Address

To omit the calling address in outgoing calls, use the **x25 suppress-calling-address** interface subcommands:

x25 suppress-calling-address
no x25 suppress-calling-address

The **suppress-calling-address** keyword omits the calling (source) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the calling address field. The calling address is sent by default.

Use the **no x25 suppress-calling-address** subcommand to reset this subcommand to the default state.

Suppressing the Called Address

To omit the called address in outgoing calls, use the **x25 suppress-called-address** interface subcommands:

x25 suppress-called-address
no x25 suppress-called-address

The **suppress-called-address** keyword omits the called (destination) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the called address field. The called address is sent by default.

Use the **no x25 suppress-called-address** subcommand to reset this subcommand to the default state.

Accepting Reverse Charge Calls

To instruct the terminal server to accept all reverse charge calls, use the **x25 accept-reverse** interface subcommands:

x25 accept-reverse
no x25 accept-reverse

The **accept-reverse** keyword causes the interface to accept reverse charge calls by default. This behavior can also be configured on a per-peer basis using the **x25 map** subcommand.

The **no** form disables this facility.

Forcing Packet-Level Restarts

To force a packet-level restart when the link level is restarted, use the **x25 linkrestart** interface subcommand:

x25 linkrestart
no x25 linkrestart

The **linkrestart** keyword restarts X.25 Level 2 (LAPB) when errors occur. This behavior is the default and is necessary for networks that expect this behavior. The **no** option disables this facility.

Setting the Packet Network Carrier

To set the packet network carrier, use the following interface subcommand:

x25 rpoa name *number*

The **x25 rpoa** interface subcommand specifies a list of transit Recognized Private Operating Agencies (RPOAs) to use, referenced by name. The argument *name* must be unique with respect to all other RPOA names. It is used in the **x25 facility** and **x25 map** interface subcommands. The argument *number* is a number that is used to describe an RPOA.

Defining a Packet Hold Queue

To define the number of packets to be held until a virtual circuit is established, use the **x25 hold-queue** interface subcommand:

x25 hold-queue *queue-size*
no x25 hold-queue [*queue-size***]**

The argument *queue-size* defines the number of packets. By default, this number is zero. Use the **no x25 hold-queue** command without an argument to remove this command from the configuration file; enter the command with a queue size value of zero to return the default.

Setting X.25 Parameters on a Per-Call Basis

To override interface settings on a per-call basis, use the **x25 facility** interface subcommand:

x25 facility *parameter value*
no x25 facility *parameter value*

The **facility** keyword enables X.25 facilities, which are sent between DTE and DCE devices to negotiate certain link parameters.

The arguments *parameter* and *value* specify one of the following:

- **cug** *number*—Specifies a Closed User Group *number* from 1 through 99 to provide an extra measure of network security.
- **packetsize** *in-size out-size*—Sets the size in bytes of input packets *(in-size)* and output packets *(out-size)*. Both values should be the same.
- **reverse**—Reverses charges on all Call Request packets from the interface.
- **windowsize** *in-size out-size*—Sets the packet count for input windows (*in-size*) and output windows (*out-size*). Both values should be the same.
- **throughput** *in out*—Sets the requested throughput values for input and output throughput across the network.
- **transit-delay** *number*—Specifies the transit delay value in milliseconds (0 to 65334) for the mapping of outgoing calls, for networks that support transit delay.
- **rpoa** *name*—Specifies the list of transit RPOAs to use in outgoing Call Request packets.

X.25 Configuration Example

The following list illustrates how to configure a terminal server interface to run DDN X.25:

Example:

```
interface serial 0
address 192.31.7.50 255.255.255.240
encapsulation DDNX25
x25 win 6
x25 wout 6
x25 ips 1024
x25 ops 1024
x25 t20 10
x25 t21 10
x25 t22 10
x25 t23 10
x25 nvc 2
x25 map IP 192.31.7.49 000000010300 BROADCAST
```

Configuring X.29 Access Lists

The terminal server software makes it possible to limit access to the terminal server from certain X.25 hosts using access lists. These access lists take advantage of the message field defined by Recommendation X.29, which describes procedures for exchanging data between two PADS or a PAD and a DTE device.

To define an item in an X.29 access list, use the **x29 access-list** configuration command. The command syntax follows:

x29 access-list *number* **permit|deny** *regular-expression*
no x29 access-list *number*

The argument *number* is a number between 1 and 99 you assign to the access list using the **access-class** line subcommand. It should be noted that the number does not determine priority of the access list, but rather is an arbitrary number assigned as a designation.

The keyword **permit** allows matching source X.121 addresses to access the terminal server. The keyword **deny** allows users to specify that their call requests clear immediately.

The X.121 address follows the **permit** or **deny** keyword. The argument *regular-expression* is the address, with or without regular expression pattern matching characters, with which to compare for access. The UNIX-style regular expression characters allow for pattern matching of characters and character strings in the address. Various pattern matching constructions are available that will allow many addresses to be matched by a single regular expression. Refer to Table 1-2 and Table 1-3 in this chapter and the appendix "Pattern Matching" for more information.

An access list can contain any number of access list items. The lists are processed in the order in which they are entered, with the first match causing the permit or deny condition, as specified. If an X.121 address does not match any of the regular expressions in the access list, access will be denied.

The **no x29 access-list** command deletes an entire access list. There is no way to delete a specific item from an X.29 access list.

Using X.29 Access Lists with Virtual Terminals

The VTYs on a terminal server can be configured with an incoming access-class using the **access-list** configuration command. But first use the **line** command as follows:

line *vty 0 99*

Then use the **access-class** command as follows:

access-class *number* **in**

The argument *number* is the number of the access list. This can be between 1 and 99.

This access list number is used for both incoming TCP access and for incoming PAD access. In the case of TCP access, the terminal server uses the IP access list defined using the **access-list** command. For incoming PAD connections, the same numbered X.29 access list is referenced. If you actually only want to have access restrictions on one of the protocols, then you can create an access list that permits all addresses for the other protocol.

Using X.29 Access Lists for Transparent Translation

Each **translate** subcommand can be configured with an access list number. In the case of translation sessions that result from incoming PAD connections, the corresponding X.29 access list is used.

X.29 Access List Example

The following is an extensive example illustrating incoming permit conditions for all IP hosts and LAT nodes with specific characters in their names and a deny condition for X.25 connections to a printer. Outgoing connections, however, are less restricted.

```
! Permit all IP hosts, LAT nodes beginning with "VMS" and no X.25
! connections  to the printer on line 5
!
access-list 1 permit 0.0.0.0 255.255.255.255
lat access-list 1 permit ^VMS.*
x29 access-list 1 deny .*
!
line 5
access-class 1 in
!
! Meanwhile, permit outgoing connections to various places on all the
! other lines.
!
! Permit IP access within cisco
access-list 2 permit 131.108.0.0 0.0.255.255
!
! Permit LAT access to the Stella/blue complexes.
lat access-list 2 permit ^STELLA$
lat access-list 2 permit ^BLUE$
!
! Permit X25 connections to infonet hosts only.
x29 access-list 2 permit ^31370
!
line 0 99
access-class 2 out
```

Creating an X.29 Profile

To create a profile, use the **x29 profile** subcommand. The command syntax follows:

x29 profile *name parameter:value* **[***parameter:value***]**

This subcommand sets up an X.3 profile for use by the **translate** command.

The argument *name* is the name assigned to the profile.

The arguments *parameter* and *value* are the X.3 PAD parameter numbers and values separated by a colon.

When an X.25 connection is established, the system acts as if an X.29 SET PARAMETER packet has been sent containing the parameters and values set by this subcommand and sets the system accordingly.

Example:

The following profile turns local edit mode on when the connection is made and establishes local echo and line termination upon receipt of a Return:

```
x29 profile linemode 2:1 3:2 15:1
```

The name *linemode* is used with the **translate** subcommand to effect use of the profile.

Refer to the chapter "Protocol Translation" for more information about the **translate** subcommand.

The X.3 PAD parameters are described in the "X.3 PAD Parameters" section earlier in this chapter.

Maintaining X.25

To clear all virtual circuits at once, use the privileged EXEC command **clear x25-vc.** This command takes an interface type keyword (usually **serial**) and a unit number as arguments to identify the interface with which the virtual circuits are associated.

To clear a particular virtual circuit, use the **clear x25-vc** command. The command syntax follows:

clear x25 vc *interface unit* **[***lcn***]**

This command clears all X.25 virtual circuits at once.

The argument *interface* is the interface type.

The argument *unit* is the interface unit number.

The optional argument *lcn* clears the specified virtual circuit.

Monitoring X.25

The terminal server provides EXEC **show** commands to provide information on interface operation and virtual circuit operation. Use the EXEC command **show interfaces** to display interface parameters and statistics. Use the EXEC command **show x25 vc** to display virtual circuit parameters and statistics.

Displaying Internet to X.121 Address Mapping

To display a mapping of Internet to X.121 addresses, use the **show x25 map** EXEC command. The command syntax follows:

show x25 map

Each line of output shows the interface name, the Internet address, the X.121 address, and the type of mapping entry. Address-mapping types are PERMANENT (entered with the **x25 map** interface subcommand), INTERFACE, indicating the address of a network interface, and Defense Data Network (DDN), derived using the DDN mapping standard. If broadcasts are enabled for an address mapping, the keyword BROADCAST also appears in the output line.

If applicable, the line also shows the count of any logical channel numbers (LCNs) and a list of the LCNs for that address. An asterisk indicates the current LCN.

The following is sample command output:

```
Serial0: 192.31.7.49 000000010300 PERMANENT, BROADCAST, 2 LCNS: 1024, 1*
Serial0: 000000020300 INTERFACE
```

Note: The output of this command is relevant only when using the Internet protocol over the X.25 interface.

Displaying PAD Parameter Information

To display information about packet transmission and X.3 PAD parameter settings, use the **show x25 pad** EXEC command. The command syntax follows:

show x25 pad

Most of the information displayed by this command will be used by an engineer to track problems.

The following is sample command output:

```
TS#show x25 pad
tty2, Incoming PAD connection
Total input: 61, control 6, bytes 129. Queued: 0 of 7 (0 bytes).
Total output: 65, control 6, bytes 696.
Flags: 1,   State: 3,   Last error: 1
 ParamsIn:  1:1, 2:0, 3:2, 4:1, 5:1, 6:0, 7:21,
    8:0, 9:0, 10:0, 11:14, 12:0, 13:0, 14:0, 15:1,
    16:127, 17:21, 18:18, 19:0, 20:0, 21:0, 22:0,
 ParamsOut:  1:1, 2:1, 3:2, 4:1, 5:0, 6:0, 7:4,
    8:0, 9:0, 10:0, 11:14, 12:0, 13:0, 14:0, 15:0,
    16:127, 17:21, 18:18, 19:0, 20:0, 21:0, 22:0,
 LCI: 1,  State: D1,  Interface: Serial0
 Started 0:11:10, last input 0:00:16, output 0:00:16
 Connected to 313700540651
```

```
Window size input: 7, output: 7
Packet size input: 512, output: 512
PS: 1  PR: 5  ACK: 5  Remote PR: 1  RCNT: 0  RNR: FALSE
Retransmits: 0  Timer (secs): 0  Reassembly (bytes): 0
Held Fragments/Packets: 0/0
Bytes 696/129 Packets 65/61 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0
```

`Total input/output` displays the number of packets received or sent for this connection. `Control` displays the number of packets with Qbit set (X.29 control packets). `Bytes` displays the number of bytes in each direction. `Queued` displays the number of unread packets waiting for this connection. `Waiting to send` displays the local data packetized bit not sent (part of a line). `Flags, state, last error` displays data useful only to Cisco, for detecting errors and tracing initialization status. `Params In` displays the parameters read from the PAD at the start of the connection. `ParamsOut` displays the active X.3 parameters. `LCI data` displays the x25 state of the connection.

The line beginning `LCI:` starts a display of the status of the X.25 virtual circuit associated with this PAD connection, and is the same display seen when the **show x25 vc** command is executed.

Displaying Active X.25 Virtual Circuit Status

To display the details of the active X.25 switched virtual circuits, use the **show x25vc** EXEC command. The command syntax follows:

show x25 vc [*lcn***]**

The optional argument *lcn* is specified to examine a particular virtual circuit. The following is sample command output:

```
LCI: 1024,  State: D1,  Interface: Serial0
Started 0:00:06, last input 0:00:06, output 0:00:06
Connected to 31370
Window size input: 7, output: 7
Packet size input: 512, output: 512
PS: 2  PR: 2  ACK: 2  Remote PR: 2  RCNT: 0  RNR: FALSE
Retransmits: 0  Timer (secs): 0  Reassembly (bytes): 0
Held Fragments/Packets: 0/0
Bytes 48/10 Packets 2/2 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0
```

Some of the command output is specific to the Cisco implementation of X.25. For example, the `Started` field shows the number of hours, minutes, and seconds since the virtual circuit was created. A `Precedent` field on the third line will appear only when you have specified DDN encapsulation, and will indicate IP precedence.

On the sixth line, the `PS` and `PR` fields show the current send and receive sequence numbers, respectively. The `Remote PR` field shows the last number received from the other end of the circuit. The `RCNT` field shows the count of unacknowledged input packets. The `RNR` field shows the state of the Receiver Not Ready flag.

On the seventh line, the `Retransmits` field shows the number of times the current packet has been retransmitted. The `Timer` field, if nonzero, shows the current value in seconds of the LCI event timer. The `Reassembly` field shows the number of bytes received for a partial packet (a packet in which the M-bit is set).

On the eighth line, the `Fragments` part of the `Held Fragments/Packets` field shows the number of X.25 packets being held. (In this case, fragment refers to the X.25 fragmentation of IP data packets.) The Packets part of the `Held Fragments/Packets` field shows the number of IP packets currently being held pending the availability of resources, such as the establishment of the virtual circuit.

On the last line, the `Bytes` field shows the number of bytes sent and received. The `Packets, Resets, RNRs, REJs,` and `INTs` fields show sent and received counts for each packet type.

Displaying Interface Routes

To display the routes assigned by the **x25 route** command, enter the **show x25 route** EXEC command.

show x25 route

Sample output follows:

```
Number            X.121              CUD      Forward To
  1    03$                                    translation, 0 uses
  2    22220102                               translation, 0 uses
  3    11110101                               translation, 0 uses
  4    ^1111                                  Serial0, 0 uses
  5    ^2222                                  Serial1, 0 uses
  6    ^3333                                  Serial2, 0 uses
  7    ^4444                                  Serial3, 0 uses
```

Debugging X.25

Use the privileged EXEC debugging commands described in this section to monitor error information about X.25 traffic. For each **debug** command, there is a corresponding **undebug** command to disable the reports.

debug lapb

To enable logging of LAPB (X.25 Level 2) transactions, use the **debug lapb** EXEC command.

debug x25

To log all the X.25 circuit activities and traffic, use the **debug x25** EXEC command:

debug x25-events

To log X.25 events, use the **debug x25-events** EXEC command.

This command enables the monitoring of all X.25 traffic but does not display information about X.25 data or acknowledgment packets. Unlike the **debug x25** command, the **debug x25-events** command does not report all data activity, thus reducing the volume of output.

Unlike the **debug x25** command, the **debug x25-events** command does not report all data activity, thereby producing terse output.

debug pad

The **debug pad** command shows debugging for all PAD connections.

debug x25-vc [*lcn*]

To log X.25 activity on virtual circuits, use the **debug x25-vc** EXEC command.

This command enables logging of X.25 activity on all virtual circuits. To log this activity for a particular virtual circuit, specify an LCN by adding the optional argument *lcn*.

The following is a sample **debug x25-vc** command output:

```
Serial0: X25 I R1 RESTART (5) 8 lci 0 cause 7 diag 250
Serial0: X25 0 R1 RESTART CONFIRMATION (3) 8 lci 0
Serial0: X25 0 P2 CALL REQUEST (19) 8 lci 1
From(14): 31250000000101 To(14): 31109090096101
Facilities (0)
Serial0: X25 0 P6 CLEAR REQUEST (5) 8 lci 1 cause diag 122
```

For each event, the first field identifies the interface on which the activity occurred, and the second field indicates that it was an X.25 event. The third field indicates whether the X.25 packet was input (I) or output (O). The fourth field is the state of the interface: R1 is the normal ready state, R2 indicates the DTE not-ready state, and R3 indicates the DCE not-ready state.

The fifth field is the type of the X.25 packet that triggered the event, and the sixth field (in parentheses) gives the total length of the X.25 packet in bytes. The seventh field is the window modulus. The eighth field (labeled `lci`) shows the virtual circuit number. The ninth field (labeled `cause`) gives the cause code, and the tenth field (labeled `diag`) gives the diagnostic code.

For Call Request and Call Connected packets, the terminal server shows additional information on separate lines. The `From` field shows the calling X.121 address and the `To` field shows the called X.121 address. The number in parentheses after the field name [`(14)`] specifies the number of digits in the address. The `Facilities` field indicates the length (in bytes) of the requested facilities and the facilities contents.

X.25 Global Configuration Command Summary

This section provides an alphabetical list of the global configuration commands.

[no] x25 route [# *position***]***x121-pattern* **[cud** *pattern***] interface** *interface-name*
[no] x25 route [# *position***]***x121-pattern* **[cud** *pattern***] ip** *ip-address*
[no] x25 route [# *position***]***x121-pattern* **[cud** *pattern***] alias** *interface-name*

Inserts or removes an entry in the X.25 routing table. The optional positional parameter (# followed by an integer) designates after which existing entry to insert or delete the new entry. No positional parameter appends the entry. The argument *x121-pattern* is the X.121 called address. The Call User Data field (**cud** *pattern*) can be entered as an ASCII string and specifies the data after the protocol identification field, which is four bytes. All pattern arguments can be entered as regular expressions. The **interface** keyword and *interface-name* argument define the interface. The **ip** keyword and **ip-address** argument specify an IP address. The **alias** keyword and *interface-name* argument permits a way for other X.121 addresses to be treated as local addresses. Enter the **no x25 route** command with the appropriate arguments and keywords to remove an entry from the table.

[no] x29 access-list *number* **permit|deny** *regular-expression*

Defines an item in an X.29 access list. The argument *number* is the number of the access list. This can be between 1 and 99. The keyword **permit** allows the source X.121 addresses matching the regular expression to access the server. The keyword **deny** allows users to specify that their call requests clear immediately. The argument *regular expression* is a regular expression similar to those used in the **translate** and **X.25-route** commands. The **no x29 access-list** command deletes an entire access list.

X.25 Interface Subcommand Summary

This section provides an alphabetical list of all the interface subcommands used to configure the X.25 interface.

encapsulation *encapsulation-type*

This subcommand takes one argument, *encapsulation-type.* For the purposes of an X.25 network configuration, this command specifies the type of X.25 support (commercial or DDN) and whether the connection is Data Terminal Equipment (DTE) or Data Communications Equipment (DCE). DDN X.25, a variant of X.25 with automatic Internet-to-X.121 address conversion, is used primarily to pass Internet traffic on the Defense Data Network. Most connections to X.25 switches are DTE.

[no] ip tcp header-compression [**passive**]

Implements header compression on IP packets.

lapb k *window-size*

Specifies the maximum permissible number of outstanding frames, called the "window size."

The argument *window-size* is a packet count from 1 to 7. The default value is 7 packets.

lapb n1 *bits*

Specifies the maximum number of bits a frame can hold. The **n1** keyword specifies the maximum number of bits (N1) a frame can hold. The argument *bits* is the number of bits from 1 through 12000, and must be a multiple of eight. The default value is 12000 bits (1500 bytes).

lapb n2 *retries*

Specifies the maximum number of times an acknowledgment frame can be retransmitted.

The argument *retries* is the retransmission count from 1 through 255. The default value is 20 retransmissions.

lapb t1 *milliseconds*

Sets the limit for the retransmission timer (the LAPB T1 parameter).

The argument *milliseconds* is the number of milliseconds from 1 through 64000. The default value is 3000 milliseconds.

[no] x25 accept-reverse

Instructs the terminal server to accept all reverse charge calls. This behavior can also be configured on a per-peer basis using the **x25 map** subcommand. The **no** variation resets the default state.

x25 address *X.121-address*

Sets the X.121 address of a particular network interface. The address is assigned by the X.25 network. The argument *X.121-address* is a variable-length X.121 address.

[no] x25 default *protocol*

Specifies or removes the protocol assumed by the terminal server to interpret calls with unknown Call User Data. The argument *protocol* sets the default protocol and is either **ip** or **pad**.

[no] x25 facility *parameter value*

Overrides interface settings on a per-call basis and enables X.25 facilities, which are sent between DTE and DCE devices to negotiate certain link parameters.

The arguments *parameter* and *value* may be as follows:

- **cug** *number*—Specifies a Closed User Group *number* from 1 through 99 to provide an extra measure of network security.
- **packetsize** *in-size out-size*—Sets the size in bytes of input packets *(in-size)* and output packets *(out-size)*. Both values should be the same.
- **reverse**—Reverses charges on all Call Request packets from the interface.
- **windowsize** *in-size out-size*—Sets the packet count for input windows (*in-size*) and output windows (*out-size*). Both values should be the same.
- **throughput** *in out*—Sets the requested throughput values for input and output throughput across the network.
- **transit-delay** *number*—Specifies the transit delay value in milliseconds (0 to 65334) for the mapping of outgoing calls, for networks that support transit delay.
- **rpoa** *name*—Specifies the list of transit RPOAs to use in outgoing Call Request packets.

x25 hic *channel*

Sets the highest incoming channel (HIC). The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

x25 hoc *channel*

Sets the highest outgoing channel (HOC). The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

[no] x25 hold-queue *queue-size*

Defines the number of packets to be held until a virtual circuit is established. The argument *queue-size* defines the number of packets. By default, this number is zero. Use the **no x25 hold-queue** command without an argument to remove this command from the configuration file; enter the command with a queue size value of zero to return the default.

x25 hold-vc-timer *minutes*

Prevents overruns on X.25 switches for traffic through the VCs. The argument *minutes* is the number of minutes by which to prevent calls to a previously failed destination. Incoming calls will still be accepted.

[no] x25 host *name 121-address* **[cud** *call-user-data***]**

Defines a static host-name-to-address mapping in the host cache. The argument *name* is the host name. The keyword **cud** sets the call user data field in the X.25 call request packet.

x25 htc *channel*

Sets the highest two-way channel (HTC). The argument *channel* is a channel number from 1 through 4095. The default value is 1024.

x25 idle *minutes*

Sets the period of inactivity once an SVC has been cleared. The argument *minutes* is the number of minutes in the period. The default value is zero, which causes the terminal server to keep the SVC open indefinitely. Both calls originated and terminated by the terminal server are cleared.

[no] x25 ip-precedence

Enables or disables the ability to open a new virtual circuit based on the IP Type of Service (TOS) field. By default, Cisco terminal servers open one virtual circuit for each type of service.

x25 ips *bytes*
x25 ops *bytes*

Set the terminal server packet size to match those of the network. The **ips** keyword specifies the terminal server input packet size while the keyword **ops** specifies the terminal server output packet size. The argument *bytes* is a byte count in the range of 128 through 1024. The default value is 128 bytes. Larger packet sizes are better, because smaller packets require more overhead processing.

x25 lic *channel*

Sets the lowest incoming channel (LIC). The argument *channel* is a channel number from 1 through 4095. The default value is 1.

[no] x25 linkrestart

Forces a packet-level restart when the link level is restarted and restarts X.25 Level 2 (LAPB) when errors occur. This behavior is the default and is necessary for networks that expect this behavior. The **no** variation resets the default state.

x25 loc *channel*

Sets the lowest outgoing channel (LOC). The argument *channel* is a channel number from 1 through 4095. The default value is 1.

x25 ltc *channel*

Sets the lowest two-way channel (LTC). The argument *channel* is a channel number from 1 through 4095. The default value is 1.

[no] x25 map *protocol-keyword protocol-address X.121-address* **[***option1 …* **[***option***]]**

Specifies a network-protocol-to-X.121 address mapping from Internet-to-X.121. For the terminal server, the argument *protocol-keyword* can only be **ip**. The *address* arguments specify the network-protocol-to-X.121 mapping. The *option* arguments add certain features to the mapping specified, and can be any of the following, up to six. (They must be specified in the order listed.)

- **reverse**—Specifies reverse charging for outgoing calls.
- **accept-reverse**—Causes the terminal server to accept incoming reverse-charged calls. If this option is not present, the terminal server clears reverse charge calls.
- **broadcast**—Causes the terminal server to direct any broadcasts sent through this interface to the specified X.121 address.
- **cug** *number*—Specifies a Closed User Group number (from 1 to 99) for the mapping in the outgoing call.
- **nvc** *count*—Sets the number of virtual circuits (VCs) for this map/host. The default *count* is the **x25 nvc** setting of the interface. A maximum number of eight VCs can be configured for a single map/host.
- **packetsize** *in-size out-size*—Specifies input packet size *in-size* and output packet size *out-size* for the mapping in the outgoing call.
- **windowsize** *in-size out-size*—Specifies input window size *in-size* and output window size *out-size* for the mapping in the outgoing call.
- **throughput** *in out*—Requests the amount of bandwidth through the X.25 network.
- **modulo** *size*—Specifies the maximum window size for this map. The argument *size* permits windows of 8 or 128 on the same interface.
- **transit-delay** *number*—Specifies the transit delay value in milliseconds (0 to 65334) for the mapping of outgoing calls, for networks that support transit delay.
- **nuid** *username password*—Specifies that a network ID facility be sent in the outgoing call with the specified user name and password.

x25 map compressedtcp *ip-address x.121-address* **[***options***]**

Makes the X.25 calls complete for compressed packets. The argument *ip-address* is the IP address and *x.121-address* is the X.121 address. The *options* arguments are the same options as those for the **x25 map** command; see above. The Call User Data of compressed TCP calls is the single byte 0xd8.

x25 modulo *modulus*

Sets the modulus. The argument *modulus* is either 8 or 128. The default value is 8. The value of the modulo parameter must agree with that of the device on the other end of the X.25 link.

x25 nvc *count*

Specifies the maximum number of switched virtual circuits that can be open simultaneously to one host. The argument *count* is a circuit count from 1 to 8; the default is 1.

x25 rpoa name *n*

Specifies a list of transit RPOAs to use, referenced by name. The argument *name* must be unique with respect to all other RPOA names. It is used in the **x25 facility** and **x25 map** interface subcommands. The argument *n* is a number that is used to describe an RPOA.

[no] x25 suppress-called-address

Omits the called (destination) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the called address field. The called address is sent by default. Use the **no x25 suppress-called-address** subcommand to reset this subcommand to the default state.

[**no**] **x25 suppress-calling-address**

Omits the calling (source) X.121 address in Call Request packets. This option is required for networks that expect only subaddresses in the calling address field. The calling address is sent by default. The **no** variation resets the default state.

x25 t10 *seconds*

Sets the limit for the Restart Request retransmission timer (T10) on DCE devices. The argument *seconds* is a time value in seconds. The default value is 60 seconds.

x25 t11 *seconds*

Sets the limit for the Call Request retransmission timer (T11) on DCE devices. The argument *seconds* is a time value in seconds. The default value is 180 seconds.

x25 t12 *seconds*

Sets the limit for the Reset Request retransmission timer (T12) on DCE devices. The argument *seconds* is a time value in seconds. The default value is 60 seconds.

x25 t13 *seconds*

Sets the limit for the Clear Request retransmission timer (T13) on DCE devices. The argument *seconds* is a time value in seconds. The default value is 60 seconds.

x25 t20 *seconds*

Sets the limit for the Restart Request retransmission timer (T20) on DTE devices. The argument *seconds* is a time value in seconds. The default is 180 seconds.

x25 t21 *seconds*

Sets the limit for the Call Request retransmission timer (T21) on DTE devices. The argument *seconds* is a time value in seconds. The default value is 200 seconds.

x25 t22 *seconds*

Sets the limit for the Reset Request retransmission timer (T22) on DTE devices. The argument *seconds* is a time value in seconds. The default value is 180 seconds.

x25 t23 *seconds*

Sets the limit for the Clear Request retransmission timer (T23) on DTE devices. The argument *seconds* is a time value in seconds. The default value is 180 seconds.

x25 th *delay*

Instructs the terminal server to send acknowledgment packets when it is not busy sending other packets, even if the number of input packets has not reached the **win** count, which improves line responsiveness at the expense of bandwidth.

The argument *delay* must be between zero and the input window size. A value of 1 sends one Receiver Ready acknowledgment per packet at all times. The default value is zero, which disables the delayed acknowledgment strategy.

x25 win *packets*
x25 wout *packets*

Set the upper limits on the number of outstanding unacknowledged packets.

The **win** keyword specifies the upper limits of the number of outstanding unacknowledged packets in the input window.

The **wout** keyword specifies the upper limits of the number of outstanding unacknowledged packets in the output window.

The argument *packets* is a packets count. The packet count for **win** and **wout** can range from 1 to the window modulus. The default value is 2 packets.

[no] x29 profile *name parameter:value* **[***parameter:value***]**

Creates an X.3 profile for use by the **translate** command. The argument *name* is the name assigned to the profile. The arguments *parameter* and *value* are the X.3 PAD parameter numbers and values separated by a colon.